A Handwriting Text Input System

This is a continuation of 08/330,573, filed on 10/28/94.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a text input system to lower the burden of text input for an information processing and communication control system.

As one of the text input method, a conventional system utilizes an abbreviation or a shorthand word. For example, it has a dictionary with plural number of words which are compared with the an input data, and the words corresponding to the input is fetched from the dictionary, and no more input is necessary.

However, in the above-mentioned method, it is necessary to remember the abbreviation and activate the system function to find a desired word by depressing a corresponding key, and its efficiency depends on an operator's capability, as the operator has to remember all the characters of abbreviation or shorthand words in the dictionary which correspond to those of input data for the successful functioning to find a match.

2. Description of the Prior Art

The conventional method has a common way to accept a leading part of word data or a radical of Chinese character or Japanese kanji character from a keyboard or a handwriting input apparatus. After an operator depresses an enter key or a space key, for example, the input data is compared with the contents of a dictionary to find the word which equates the input data and display, if any. The operator may find desired words on the display and select one of them by the depression of a specific function key.

However, the method to press the enter key or the space key as a separator after typing some characters to find the desired word is not suitable for typing on the keyboard in a blind (eyes-free)

manner. It is also unlikely that the operator reiterates the steps to input a line of text, character by character, for the system to collate with the contents of dictionary and display word data with same leading part as the input, until some of them is selected by the operator. Its method heavily relies on the operator's ability and does not necessarily contribute to the increase in input efficiency.

In the case of handwriting input, there would be plenty of words in the dictionary with the same radical as those of Chinese or Japanese kanji characters which will cause a frequent change of a list of words for the selection on the display after some data input and repeated steps to select the desired word among them. Such action may actually decrease the efficiency of data input, eg. the system first displays the list of radicals and the operator selects one of the radicals to which the system gets back to the operator with the relevant Chinese or Japanise kanji characters for the selection on the display by the operator.

SUMMARY OF THE INVENTION

The text input system of the invention, was intended to provide a way of data input from a keyboard or a handwriting input apparatus, giving an efficient and natural way of input for an operator without being conscious of it while using and by the step equivalent to what they have been doing up until now.

The text input method should not give an extra burden and an additional training to the operator, and should be able to increase an efficiency of the data input using a natural way of input, as well.

In comparison with the conventional method, the invention system collates an input data with a dictionary, character by character, at the time of each data input and replaces the data input on the display with the unique one from the dictionary after finding it which includes data input in a variety of forms, and requires no more input. This increase the efficiency of data input

and lower the burden on operator as well as a variety of data input method mentioned below.

Additionally, this invention is a flexible and effective way to further increase the efficiency of input, by decreasing the number of words in the dictionary containing the words only with the number of characters more than the specific number, in the dictionary, by way of storing relevant words together with the words to collate with the data input, in the dictionary. It decreases the amount of data input and shorten the time required to reach the character position which makes the word unique in the dictionary for the purpose of supplementing the remaining part of word from the dictionary.

In the abbreviation input, the conventional method uses a dictionary consisting of an abbreviation and its original form of word, though the present invention is able to use a standard dictionary, as well as the abbreviation dictionary.

Systems, according to the invention, do not require a special abbreviation dictionary. It is able to utilize the standard dictionary and accept the abbreviation input which contains the first character and some other following characters or strokes of the word data to input and compares those with the dictionary. Even in case of having the special abbreviation dictionary, the present invention does not require all the characters or strokes of the abbreviation to input. In this case, the operator does not have to remember the abbreviation and may input part of it, eg. the first character followed by some other characters of the abbreviation which is unique in the dictionary.

The present invention has a feature to enter a line of text, character by character, collate the entered line of text with a dictionary storing a plurality of lines of text and relevant words, determine a unique line of text in the dictionary which includes the entered line of text, and select a unique word among relevants words, at the time of character input, without being actuated by the depression of a special function key, and replace the entered line

of text with the unique line of text or the unique word in the dictionary.

The present invention has a feature to enter a line of text, character by character, collate the entered line of text with a dictionary, identify plural lines of text with the same stem of word which includes said entered line of text, and determine a unique line of text which has the same last character as the last entered character, among said identified plural lines of text, without being actuated by the depression of a special function key, at the time of character input, and also identify plural lines of text with the same stem of word which includes said entered line of text, and determine said unique line of text which includes the same one as the last entered characters in the remaining part of line of text in the dictionary other than that was successfully collated with entered line of text, among said identified plural lines of text, wihtout being actuated by the depression of a special function key, at the time of character input.

The present invention has a feature to enter a line of text, character by character, collate the entered line of text with a dictionary, identify plural lines of text with the same first part as said entered line of text, and determine a unique line of text which has the same last character as the last entered character, among said identified plural lines of text, without being actuated by the depression of a special function key, at the time of character input, and also identify plural lines of text with the same first part which includes said entered line of text, and determine a unique line of text which includes the same one as the last entered characters in the remaining part of line of text in the dictionary other than that was successfully collated with entered line of text,, among said identified plural lines of text, without being actuated by the depression of a special function key, at the time of character input.

The present invention has a feature to enter a first character followed by some other following characters of a line of text to variably input, character by character, collate entered line

of text with a dictionary, determine a unique line of text in the dictionary which includes the first character and some other following characters of line of text, at the time of character input, without being actuated by the depression of a special function key.

The present invention has a feature to enter a line of text, character by character, collate entered line of text with a dictionary storing a plurality of lines of text, original words, and a unique position count, and determine a unique line of text which has the same unique position count as the number of last collated character position of line of text in the dictionary collated with the entered line of text, without being actuated by the depression of a special function key, at the time of following character input.

The present invention has a feature to enter a line of text, character by character, collate entered line of text with a dictionary storing a plurality of lines of text, determine a unique line of text which includes entered line of text, without being actuated by the depression of a special function key, at the time of following character input.

The present invention has a feature to enter a line of text of handwriting strokes, stroke by stroke, collate the entered line of text with a dictionary storing a plurality of lines of text of handwriting strokes, original words and a unique position count, determine a unique line of text of handwriting strokes in said dictionary which includes said entered line of text of handwriting strokes and which has the unique position count same as the number of last collated stroke position of the line of text of handwriting strokes in said dictionary collated with said entered line of text of handwriting strokes, at the time of entering the handwriting strokes, and replace the entered line of text with the unique line of text or original word in the dictionary.

The present invention has a feature to enter a line of text of handwriting strokes, stroke by stroke, collate entered line of text of handwriting strokes with a dictionary, identify plural lines of text of handwriting strokes with the same first part as said entered line of text, and determine a unique line of text of

handwriting strokes which has the same last stroke as the last entered stroke, among identified plural lines of text, without being actuated by the depression of a special function key, at the time of stroke input, and also identify plural lines of text of handwriting strokes with the same first part which includes entered line of text, and determine a unique line of text of handwriting strokes which includes the same one as the last entered strokes in the remaining part of line of text of handwriting strokes in the dictionary other than that was successfully collated with entered line of text, among said identified plural lines of text of handwriting strokes, without being actuated by the depression of a special function key, at the time of stroke input.

The present invention has a feature to enter a first stroke followed by some other following strokes of line of text of handwriting strokes to variably input, stroke by stroke, collate entered line of text with a dictionary, determine a unique line of text of handwriting strokes in the dictionary which includes the first stroke and some other following strokes of line of text of handwriting strokes, at the time of stroke input, without being actuated by the depression of a special function key, and replace the entered line of text with the unique line of text in the dictionary.

The present invention may comprise means for determining a predetermined number range of lines of text in said dictionary.

The present invention may comprise means for storing said lines of text in said dictionary which is organized in a random access manner.

Some examples for the number of character positions in a word to be unique in the dictionary in comparison with the total number of characters of the word, are given in Table 2. The examples of a line of text of characters or strokes are shown in table 1.

Also, there is a possibility to be more quick in finding the unique word in the dictionary, by decreasing the number of short words. For example, "abolish" can be identified at the 4th character position (ie. abol), in the selective dictionary of the following

table 2, though it needs 6 characters to be unique in this standard dictionary.

- Table 1 -
(An example of dictionary)

Word pattern element codes for handwriting strokes	Original word	The number of position to be unique in the dictionary
02 01 12 11 02	正	2
02 11 25 12 12 00 30	更	2
00 30 12	A	1
01 34	B	1
12 01 11 11	E	1

* In this simple examples, "正" or "更" can be unique at the second pattern element code position.

- Table 2 -

Word (or line of text)	Standard Dictionary: The number of character position in a word to be unique in the dictionary	Standard Dictionary: The total number of char. of the word	Selective Dictionary: The number of character position in a word to be unique in the dictionary	
abandon	4	7	abandon	3
abate	4	5		
abbot	4	5		
abdomen	3	7	abdomen	3
abhor	3	5		
abide	4	5		
ability	4	7	ability	3
abject	3	6		
able	4	4		
abnormal	3	8	abnormal	3
aboard	4	6		
abolish	6	7	abolish	4
abolition	6	9		
abominable	4	10		
abound	5	6		
about	5	5	about	4
above	4	5		
abridge	4	7	abridge	3
abroad	4	6		
abrupt	4	6		

absence	6	7	absence	4
absent	6	6		
absinthe	4	8		
absolute	6	8	absolute	5
absolve	6	7		
absorb	5	6		
absorbent	7	9	absorbent	5
abstain	5	7	abstain	4
(Total)	127	184	41	79

(127 / 184 = 0.69 : 31% differences

(41 / 79 = 0.52 : 48% differences (79 is the total number of char.)

In the drawings:

Embodiments of the invention will now be described with reference to the drawings in which; Fig. 1 is a block diagram of the invention system. Figures 2 to 11 show the functional specifications of the present invention; and Figures 12 to 14 show examples of the dictionary to be used by the invention system.

In the flowchart of Fig. 2 to Fig. 11, A001 is the first step to clear the contents of counters, flags, and work area, at the beginning of this data input process.

Next A002 accepts a data input, character by character, or stroke by stroke, from a keyboard or a handwriting input instruments.

A003 checks if the input is an END code to show the end of transaction, and if yes, the step goes to the termination of this process. Otherwise, the step proceeds to A003A.

A003A tests if the NONE flag is ON which means that there is no data including the data input in the dictionary. If YES, the step

goes to S020 in which the ID check steps are performed to find the unique one with the last data same as the last input data in the remaining part of line of text in the dictionary other than that was already collated (in Figure 9), as the ID check steps are performed after the input of last data and prior to its dictionary search, utilizing a flag of NONE flag which is set ON at A006B in Figure 3. It means that the attempt to search at A006A in Figure 3 using the previous input data has failed and the NONE flag was set ON for the purpose to activate ID check steps to find the data for "a leading data and some other following data to be unique in the dictionary." If None flag is off, the step proceeds to next A003B.

A003B tests if the input data is a separator code. If it is not, the process proceeds to A004 in which input data is displayed at the end of data input. If it is the separator code at the testing of A003B, the step goes to A003C in which the input buffer is cleared in this case and the step goes back to A002 for the next data input.

After displaying the input data at A004, the next A005 adds the input to the end of previous data string in the input buffer memory. Then, the step goes to A006 that is a dictionary search.

A006 performs the dictionary search to get data which has the same leading part as the data input and which is unique in the dictionary, as one of example in the case, because there are some other cases to find the data which is unique, despite the different form of data, eg. data input of the first data followed by some other following data for the line of text data to represent the unique one in the dictionary, and others.

Next A006A tests if the result of dictionary search at A006 shows that there are data to retrieve in this example. If there are no data, the step goes to A006B in which NONE flag is turned on and goes back to A002 for the next data input. In this example, NONE flag is used for the purpose to do a sequence of "input of following data and ID check" to find the unique data with the same last data as the last data input or with the same last data input in the remaining part of line of text in the dictionary other than that was

already collated. If the test result at A006A is not negative which means that there are data to retrieve, next A007 tests if the step of A006 found the unique one in the dictionary by testing "flag for the unique ON." If OFF, the step goes back to A002 for the next data input to continue the input and dictionary search.

At A008 branching from A007, the system replaces input data on the display with the unique data from the dictionary, and the contents of the input buffer is cleared for the next data input, and goes to next A008A.

At A008A, the selection step for the case having plural number of relevant words in the dictionary after finding the unique data in the dictionary takes place. One of relevant data is selected on the display here in this case. Other selection steps are performed by ID check at S020 in Figure 9, testing the existence of data which is unique and terminates with the same one as the last data input or which is unique and includes the same one as the last data input in the remaining part of line of text in the dictionary that was already collated. Then the step goes back to A002 for the next data input.

S001 is an entry of the subprogram to search in the dictionary. S002 is to test if there are no more word to retrieve in the dictionary. If so, the process goes to the exit of this process. If there are words to retrieve at the testing of S002, the process goes to the S003.

S003 gets the data located in the middle of the area between an upper limit and a lower limit in the dictionary. In this case, the upper limit means the boundary of data area in which its data key (value) is usually getting smaller, and the lower limit means the boundary of data area in which its data key (value) is usually getting larger, if those data are stored in an ascending order.

S004 compares data input with data from the dictionary, and tests if the leading part of data from the dictionary is greater or not. If the greater flag is ON at S004, S005 divides the retrieval area size into 2 to use the lower half area for the next retrieval, as there is a possibility to find the unique word in the lower half

area, because the dictionary data found at S003 was greater than the data input. Then, the step goes back to S002. If the greater than flag is OFF at S004, the step proceeds to S006.

At S006, it is tested if the leading part of data from the dictionary is equal to the data input. If equal, S007 is performed to test if it is single. In case of a single one, the next step S008 turns on the flag for the unique data in the dictionary and goes to exit. If there are plural number of data at S007, the step goes to S012. If NOT EQUAL at S006, the step proceeds to S009.

At S009, the testing is made if the leading part of data from the dictionary is less than that of data input.

If the less than flag is ON after testing at S009, S010 divides the retrieval area size into 2 to use the upper half area for the next retrieval, as there is a possibility to find the unique one in the upper half area, because the dictionary data found at S003 was less than that of data input. Then, the step goes to S002.

If the less than flag is OFF at S009, there is something wrong with the procedure in the process and goes to the error procedure of S011 which is not described here, due to the matter which is not directly related to the invention.

Branching here from S007, S012 tries to see if plural number of data have the same stem of word. If it is YES, the step goes to S012A in which a NONE flag is turned ON and the step goes back to A002 for the next data input. Otherwise, the step goes to S012B.

S012B gets the data with a leading part greater than that of input data in the upper half of current retrieval area in the dictionary. At the test of next S013 after performing the step of S012B, it is tested if the data was found, and goes to S015 if found. If the data was not found, the step goes to S014.

S015 sets new upper limit to retrieve using the positions of data with the leading part of data greater than that of data input, in the dictionary, and goes to next S016.

S014 sets new upper limit to retrieve in the dictionary using the position of the upper limit by which S012B was attempted, because the upper half area in this case was occupied by the words

with the leading part equal to that of data input as a result of the attempt at S012B. Then, the step goes to S016.

S016 tries to find the data with the leading part less than that of data input in the lower half area of the dictionary.

Next S017 tests if the data was found.

S018 takes place if not found at S017 and sets new lower limit using the position of the lower limit by which S016 was attempted, because the lower area was occupied by the data with the leading part equal to that of data input as a result of attempt at S016. Then, the step goes to the exit of this subprogram.

S019 arises if the data was found at S017 and sets new lower limit to retrieve using the positions of the data with the leading part less than that of the data input and jumps to the exit of this subprogram.

Branching from A003A when the NONE flag is ON, S020 performs the steps named ID check in this case to start with S023 which is an entry of subprogram. S023 and followings check if there is the unique data which has the last data same as the last data input, or which includes the same one as the last data input in the remaining part of line of text in the dictionary other than that was already collated with the data input.

Next S021 checks the flag for the unique data, after coming back from the above-mentioned ID check. If its flag is OFF, the step goes back to A002 for the next data input. If the flag is ON, the step proceeds to next S022 which turns the NONE flag off and supplements the remaining part of data input on the display, ie. the replacement of data input with the unique data found. Then, the step goes back to A002 for the next data input.

S023 is the start of subprogram which performs determination process of the unique data. S023 examines the data which were selected by the prior look-up in the dictionary.

S024 tests if there is the unique data which has a last one with same one as the last data input. If YES, the step goes to next S025 which turns the FLAG for the unique data on and goes to exit of this subprogram. If NO at the test of S024, the step goes to S026.

At S026, it is tested if there is the unique data which includes the last data input in the remaining part of line of text in the dictionary other than that was already collated with data input. If YES, the step goes to S025. If NO, the step proceeds to the next S027 in which the system extracts data which do not have the same one as the last data input or which do not include the same one as the last data input in the remaining part of line of text in the dictionary other than that was already collated with the data input, for the next retrieval.

As a subset of the dictionary search for the data input consisting of a first character and some other characters, T001 which is equivalent to the aforementioned S001, is an entry of this subprogram.

T002 which is equivalent to the aforementioned S002 checks if there are words to retrieve in the dictionary. If there are no more words, the step goes to the exit. The process advances to the next T002A, if there are words to retrieve.

T002A branches to the aforementioned S003 if the input is the first one of data input, and goes to the next T002B if it was not the first one.

T002B is to check if all words between upper and lower limit in the dictionary have the same leading part as that of data input.

T002C selects the data with the leading part which is not the same as that of data input.

T002D shifts one character to the left for every data words selected at T002C and extract those which terminated after making a character shift.

T002E sets new upper and lower limits to retrieve in the dictionary after extracting data words terminated at T002D. Then, the step goes to the aforementioned S003.

Fig. 12 is an example for the word pattern element data for the handwriting characters. In this example, let us assume that "正" is going to be input. At the time of input of its first stroke of "－", #3 and #4 words exist, having "－". Next stroke of "｜"

selectes #3 only which corresponds to the input of "-| " in this example and " 正 " is replaced with the data input on the display.

Fig. 13 and 14 are examples of the dictionary containing a line of text, original word, relevant words, and a unique position count.